

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-299411
(43)Date of publication of application : 22.10.1992

(51)Int.Cl.

G06F 5/00
G06F 12/00
H03M 7/40
H04L 23/00

(21)Application number : 03-085887
(22)Date of filing : 27.03.1991

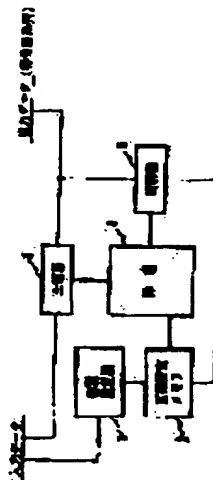
(71)Applicant : NEC CORP
(72)Inventor : OKAMURA TOSHIHIKO

(54) FILE COMPRESSOR AND FILE EXPANDER

(57)Abstract:

PURPOSE: To improve the file compression factor of a file compressor by initializing a dictionary from a file name identifier and deciding the limit value.

CONSTITUTION: The limit value of the series number registered in a dictionary 3 and the initialization of the dictionary 3 are previously stored in an initialization memory 2 in accordance with an identifier. An initialization unit 1 reads the identifier of a file to be compressed and reads out the corresponding record in the memory 2. Then the unit 1 initializes the dictionary 3 and sends the limit value of the series number stored in the dictionary 3 to a controller 5. When a coding operation is carried on and the dictionary 3 is filled up, controller 5 decides an item to be deleted out of the dictionary 3 and carries on the coding operation with repetition of the deleting/registering operations.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998.2000 Japanese Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平4-299411

(43) 公開日 平成4年(1992)10月22日

(51) Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 5/00		H 9189-5B		
	12/00	5 1 1 8944-5B		
H 0 3 M 7/40		8836-5J		
H 0 4 L 23/00		A 8020-5K		

審査請求 未請求 請求項の数6(全 6 頁)

(21) 出願番号 特願平3-85887

(22) 出願日 平成3年(1991)3月27日

(71) 出願人 000004237

日本電気株式会社
東京都港区芝五丁目7番1号

(72) 発明者 岡村 利彦

東京都港区芝五丁目7番1号 日本電気株式会社内

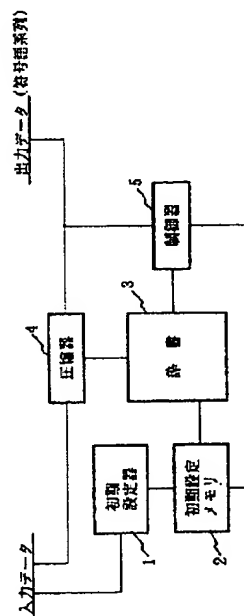
(74) 代理人 弁理士 岩佐 義幸

(54) 【発明の名称】 ファイル圧縮装置およびファイル伸張装置

(57) 【要約】

【目的】 ファイル圧縮装置において、ファイル名の識別子から辞書の初期設定、辞書の大きさを決定し圧縮率を向上させる。

【構成】 初期設定メモリ2に、予め識別子に対応して、辞書3に登録される系列の数の限界値と辞書3の初期設定を記憶させておく。初期設定器1は、圧縮するファイルの識別子を読み込み、初期設定メモリ2の対応するレコードを読み出し、辞書3を初期化し、制御器5に辞書3に登録される系列の数の限界値を伝える。符合化が進み、辞書3がいっぱいになったら、制御器5で辞書3の項目のうちで削除するものを決定し、削除・登録を繰り返して符合化を続行する。



【特許請求の範囲】

【請求項1】入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系列を前記辞書に登録してゆき、辞書に登録される入力系列の部分系列の数に制限を加え、辞書が一杯になったら削除・登録を繰り返すファイル圧縮装置において、辞書に登録される入力系列の部分系列の数の限界値をファイル名の識別子に対応して予め設定しておき、符号化開始時には識別子から前記限界値を決定することを特徴とするファイル圧縮装置。

【請求項2】入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系列を前記辞書に登録してゆき、辞書に登録される入力系列の部分系列の数に制限を加え、辞書が一杯になったら削除・登録を繰り返すファイル圧縮装置において、辞書の初期設定をファイル名の識別子に対応させて予め用意しておき、辞書はこの識別子に対応した初期設定で符号化を開始することを特徴とするファイル圧縮装置。

【請求項3】入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系列を前記辞書に登録してゆき、辞書に登録される入力系列の部分系列の数に制限を加え、辞書が一杯になったら削除・登録を繰り返すファイル圧縮装置において、辞書に登録される部分系列の数の限界値と辞書の初期設定をファイル名の識別子に対応して予め設定しておき、符号化開始時にはこの識別子から前記限界値を決定し、さらに辞書はこの識別子に対応した初期設定とすることを特徴とするファイル圧縮装置。

【請求項4】請求項1記載のファイル圧縮装置により圧縮されたデータを伸張するファイル伸張装置において、識別子に対応した辞書に登録される系列の数の限界値を、辞書に登録される入力系列の部分系列の数の限界値をファイル名の識別子に対応して予め設定しておき、符号化開始時には識別子から前記限界値を決定することを特徴とするファイル伸張装置。

【請求項5】請求項2記載のファイル圧縮装置により圧縮されたデータを伸張するファイル伸張装置において、ファイル名の識別子に対応した辞書の初期設定を、ファイル名の識別子に対応させて予め用意しておき、辞書はこの識別子に対応した初期設定で符号化を開始することを特徴とするファイル伸張装置。

【請求項6】請求項3記載のファイル圧縮装置により圧縮されたデータを伸張するファイル伸張装置において、ファイル名の識別子に対応した辞書に登録される系列の数の限界値、辞書の初期設定を、辞書に登録される部分系列の数の限界値と辞書の初期設定をファイル名の識別子に対応して予め設定しておき、符号化開始時にはこの

識別子から前記限界値を決定し、さらに辞書はこの識別子に対応した初期設定とすることを特徴とするファイル伸張装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系列を辞書に登録してゆく無歪データ圧縮装置に関する。

【0002】

【従来技術】辞書を使ったデータ圧縮装置の場合、辞書はデータの性質に拘らずメモリが許す限り大きくしてゆく方法がほとんどであった。また、データの性質により、メモリの大きさに拘らず、辞書の大きさを制限する方法であって、圧縮過程における圧縮率から辞書に登録できる部分系列の数の限界値を決めるというものであり、その限界値は圧縮過程において動的に変わるものであった。

【0003】

【発明が解決しようとする課題】データ圧縮に関しては、データによって辞書が一杯になったときには、辞書への登録を削除と交互に行う方式を用いると、辞書の大きさを小さく保ったほうが圧縮率の向上がみられることがわかっている。また、圧縮率から辞書に登録できる部分系列の数の限界値を動的に変える方法では、最適なサイズの辞書を用いた場合にわずかに劣る。また、辞書を使った圧縮方式は、立ち上がりの悪さが問題であった。つまり、符号化し始めにおける圧縮率の劣化が甚だしく、短いファイルだと圧縮をかけると逆に膨張してしまうことがよくあるのである。

【0004】本発明の目的は、このような欠点を除去し、ファイル名の識別子により辞書の設定を決定することにより、広範囲のデータに対して圧縮率の向上を見ることが出来るファイル圧縮装置およびファイル伸張装置を提供することにある。

【0005】

【課題を解決するための手段】第1の発明は、入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系列を前記辞書に登録してゆき、辞書に登録される入力系列の部分系列の数の制限を加え、辞書が一杯になったら削除・登録を繰り返すファイル圧縮装置において、辞書に登録される入力系列の部分系列の数の限界値をファイル名の識別子に対応して予め設定しておき、符号化開始時には識別子から前記限界値を決定することを特徴としている。

【0006】第2の発明は、入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系

列を前記辞書に登録してゆき、辞書に登録される入力系列の部分系列の数の制限を加え、辞書が一杯になったら削除・登録を繰り返すファイル圧縮装置において、辞書の初期設定をファイル名の識別子に対応させて予め用意しておき、辞書はこの識別子に対応した初期設定で符号化を開始することを特徴としている。

【0007】第3の発明は、入力系列の部分系列に参照番号を割り振った辞書を作り、同じ部分系列が再び現れたらその参照番号を出力し、その部分系列を延ばした系列を前記辞書に登録してゆき、辞書に登録される入力系列の部分系列の数の制限を加え、辞書が一杯になったら削除・登録を繰り返すファイル圧縮装置において、辞書に登録される部分系列の数の限界値と辞書の初期設定をファイル名の識別子に対応して予め設定しておき、符号化開始時にはこの識別子から前記限界値を決定し、さらに辞書はこの識別子に対応した初期設定とすることを特徴としている。

【0008】第4の発明は、第1の発明のファイル圧縮装置により圧縮されたデータを伸張するファイル伸張装置において、識別子に対応した辞書に登録される系列の数の限界値を、辞書に登録される入力系列の部分系列の数の限界値をファイル名の識別子に対応して予め設定しておき、符号化開始時には識別子から前記限界値を決定することを特徴としている。

【0009】第5の発明は、第2の発明のファイル圧縮装置により圧縮されたデータを伸張するファイル伸張装置において、ファイル名の識別子に対応した辞書の初期設定を、ファイル名の識別子に対応させて予め用意しておき、辞書はこの識別子に対応した初期設定で符号化を開始することを特徴としている。

【0010】第6の発明は、第3の発明のファイル圧縮装置により圧縮されたデータを伸張するファイル伸張装置において、ファイル名の識別子に対応した辞書に登録される系列の数の限界値、辞書の初期設定を、辞書に登録される部分系列の数の限界値と辞書の初期設定をファイル名の識別子に対応して予め設定しておき、符号化開始時にはこの識別子から前記限界値を決定し、さらに辞書はこの識別子に対応した初期設定とすることを特徴としている。

【0011】

【作用】いくつかの識別子に対して辞書に登録される部分系列の限界値をどのくらいに設定すればよいのか決定できるため、特に辞書を小さく保っている方がよい場合のファイルに対して圧縮率の向上を見ることが出来る。また、辞書の初期設定をその識別子に合ったものに設定することにより、符号化しはじめでの圧縮率の改善、特に短いファイルに対しての圧縮率の向上を見ることが出来る。

【0012】

【実施例】次に、本発明の実施例について、図面を参照

して説明する。

【0013】図1は、本発明の圧縮過程に係るファイル圧縮装置の一例を示すブロック図である。図1のファイル圧縮装置は、初期設定器1と、初期設定メモリ2と、辞書3と、圧縮器4と、制御器5とを備えている。

【0014】初期設定器1は、圧縮するファイルの識別子を読み込み、初期設定メモリ2の対応するレコードを読み出し、辞書3を初期化し、制御器5に辞書3に登録される系列の数の限界値を伝える。すなわち、初期設定器1は、圧縮するファイルの識別子から初期設定メモリ2のどのデータで辞書を初期化するのかを決定する。

【0015】初期設定メモリ2は、予め識別子に対応して、辞書3に登録される系列の数の限界値と辞書3の初期設定を記憶する。

【0016】辞書3は、入力系列の部分系列を蓄えるものである。

【0017】圧縮器4は、入力系列と辞書3に登録されている系列の間のマッチングをとり、出力（符号語）を決定し、また、辞書3に登録する系列を決定する。

【0018】制御器5は、符号化が進み辞書3がいっぱいになったら、辞書3の項目のうちで削除するものを決定する。すなわち、制御器5は、初期設定器の情報により辞書に登録される系列の数の限界値を設定し、辞書がいっぱいになったら、削除の位置を決定する。

【0019】次に、このファイル圧縮装置の動作について説明する。

【0020】圧縮アルゴリズムはレンベル・ジブ・ウェルチの方法に従うとし、辞書がいっぱいになったら削除・登録を行うのであるが、辞書の削除の方式は古くに現れた部分系列から順に削除してゆくとする。これは、圧縮過程における圧縮率から辞書に登録できる部分系列の数の限界値を決めるという従来の方式に記されているアルゴリズムで実現できる。これをアルゴリズムAとする。頻繁に現れるいくつかの識別子を事前に登録しておく。その識別子に対して辞書3に登録される部分系列の数の限界値と辞書の初期状態の設定を初期設定メモリ2に保存しておく。

【0021】符号化開始時には、初期設定器1により、ファイル名の識別子を読みとり、その識別子が予め登録されている識別子であったら、それに対応する設定を初期設定メモリ2から読みだし、辞書3を初期状態にし、辞書3に登録される部分系列の数の限界値を設定し圧縮を開始する。もし、登録されていない識別子であったら、例えば、圧縮過程における圧縮率から辞書に登録できる部分系列の数の限界値を決めるという従来の方式に記載の辞書サイズを可変にする方式を用いて圧縮を行う。

【0022】圧縮器4は、入力系列と辞書3に登録されている系列の間のマッチングをとり、出力（符号語）を決定し、また辞書3に登録する系列を決定する。制御器5

は、初期設定メモリ2からの情報により、辞書3に登録される系列の数の上限を設定し、辞書3が一杯になった場合には削除の位置を決定する。辞書サイズ可変方式の場合は、辞書に登録される系列の数の限界値の変更を決定するための圧縮率も計算する。圧縮前のファイルの識別子がわかるように圧縮後のファイルに名前を付ければ、伸張側でも圧縮の時と同一の初期設定メモリを用意しておくことにより、識別子から辞書の初期設定を決めることができ、符合化時と同一の辞書を作成することができるので正しい復号が行われる。

【0023】図2は、本発明の伸張過程に係るファイル伸張装置の一例を示すブロック図である。

【0024】図2において、1の初期設定器、2の初期設定メモリ、3の辞書、5の制御器については図1と同一である。6の復号器は、入力データ（符号語系列）から符号語を読みとり、辞書3の対応するレコードにあたることにより出力系列を生成し、また辞書3に登録する系列を決定する。

【0025】辞書の初期設定については、普通レンベル*

*・ジブ・ウェルチの方式では入力アルファベット全体に登録するだけであるが、本発明では、例えば、よく現れるフレーズを初期設定メモリ2に保存しておくのである。例えば、識別子が“.C"で表されるファイル、つまりCのプログラムファイルの場合、入力アルファベットをbyte単位として、参照番号256を制御用（データ終了）用コードにとっておくとすると、参照番号257から303までに図3のような系列を予め登録しておくといよい。なお、図3において、△はスペース（アスキーコード32）を表し、*nは改行（アスキーコード10）を表す。

【0026】辞書に登録される系列の数の限界値を510、1022、2046、4094、8190と5種類用意してアルゴリズムAを用いていくつかのファイルを圧縮した結果を表1に示す。

【0027】

【表1】

【0028】

	510	1022	2046	4094	8190
*.C (52154)	22953(.440)	19062(.355)	17629(.331)	16486(.316)	16460(.318)
*.C (51703)	28327(.548)	24271(.489)	22211(.430)	20398(.395)	20394(.394)
*.BAS (46466)	20240(.724)	17567(.629)	17567(.629)	16645(.596)	16559(.593)
*.EXE (27632)	21150(.760)	21501(.773)	22097(.794)	22748(.817)	23459(.843)
*.EXE (40401)	23598(.584)	23191(.574)	23598(.587)	24278(.601)	24946(.617)
*.SYS (20756)	13480(.649)	13527(.652)	13941(.672)	14414(.694)	14707(.709)

【0029】すなわち、表1は、辞書に登録される系列の数の限界値を510、1022、2046、4094、8190の5種類に設定しそれぞれで、ファイルを圧縮した結果の表である。表1から分かるように、辞書に登録される系列の数の限界値は“.SYS"、".EXE"などの識別子のファイルに対しては小さく設定しておくといよい。逆に".C"、".BAS"などのプログラムファイルや文書ファイルなどに対してはメモリを最大限利用して辞書を大きくしていった方がよい。

【0030】表1から、".SYS"、".EXE"などのファイルは、辞書に登録される部分系列の数の限界

値を510に設定しておけば、限界値が8190の時と比較して、元のファイルが20000～40000bytesの場合で5～10%圧縮率が向上することがわかる。また、図3における".C"の場合の初期設定によってCの短めのプログラムを圧縮した場合と通常の初期設定（入力アルファベットのみを登録しておく）で圧縮した場合の結果を、表2にまとめておく。

【0031】

【表2】

【0032】

	通常の初期設定	今回の設定
FILE1(2702bytes)	2279 (84%)	1409 (52%)
FILE2(825bytes)	812 (98%)	563 (69%)
FILE3(3415bytes)	2845 (83%)	1786 (52%)
FILE4(2812bytes)	2440 (87%)	1490 (53%)

【0033】表2に示されるように、圧縮率において30%程度の差が生まることがわかる。

【0034】

【発明の効果】以上説明したように、本発明は、ファイル圧縮において、ファイル名の識別子から辞書の初期設定、辞書の大きさを決定し、圧縮率を向上させる効果がある。

【図面の簡単な説明】

【図1】本発明の圧縮過程に係るファイル圧縮装置の一 20 例を示すブロック図である。

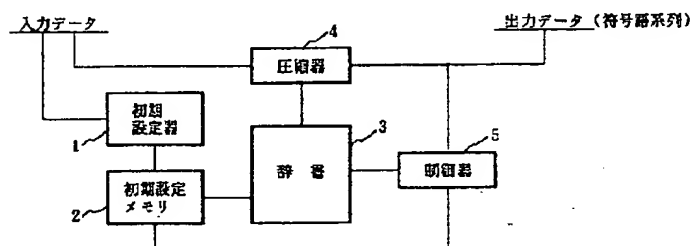
【図2】本発明の伸張過程に係るファイル伸張装置の一 例を示すブロック図である。

【図3】辞書の初期設定の一例を示す図である。

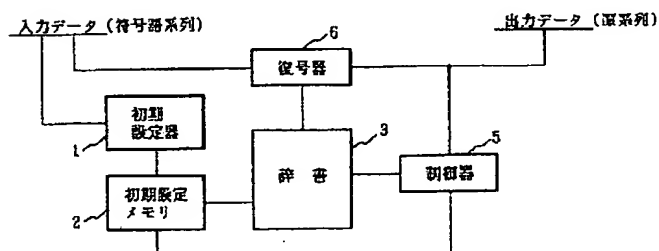
【符号の説明】

- 1 初期設定器
- 2 初期設定メモリ
- 3 辞書
- 4 圧縮器
- 5 制御器

【図1】



【図2】



(6)

特開平4-299411

【図3】

参照番号	系列	参照番号	系列
257	△ △	279	()
258	△△ △	280	() ;
259	△△△ △	281	() : Va
260	△△△△ △	282	f o
261	△△△△△ △	283	fo r
262	△△△△△△ △	284	w h
263	△△△△△△△ △	285	wh i
264	};	286	whl l
265	}; Vn	287	whil e
266	}; Vn △	288	i n
267	}; Vn△ △	289	in t
268	}; Va△△ △	290	int △
269	}; Vn△△△ △	291	c h
270	; Vn	292	ch a
271	:Vo △	293	cha r
272	:Vn△ △	294	char △
273	:Vn△△ △	295	△ =
274	:Vn△△△ △	296	△ = △
275) ;	297	p r
276); Vn	298	pr l
277	i f.	299	pr n
278	if △	300	prin t
		301	print f
		302	printf (
		303	printf("